

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



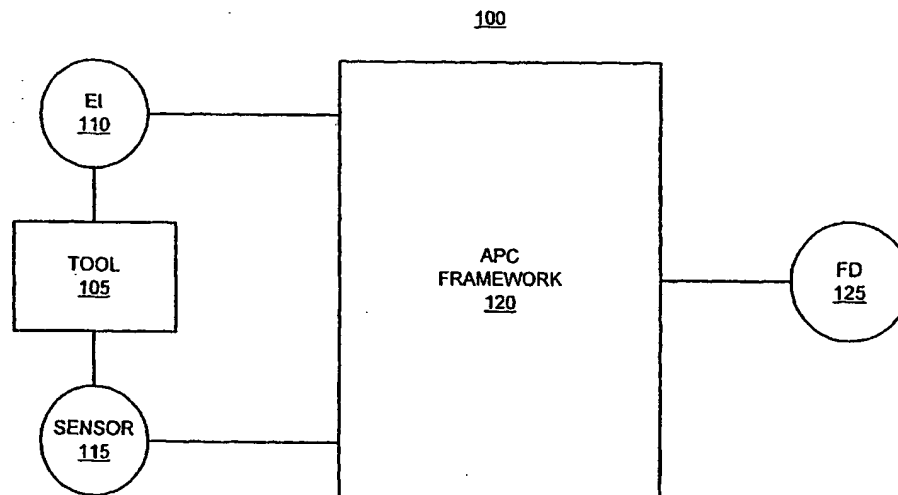
(43) International Publication Date  
15 March 2001 (15.03.2001)

PCT

(10) International Publication Number  
**WO 01/18623 A1**

- (51) International Patent Classification<sup>7</sup>: **G05B 19/418** (74) Agent: **DRAKE, Paul, S.**; Advanced Micro Devices, Inc.,  
5204 East Ben White Boulevard, M/S 562, Austin, TX  
78741 (US).
- (21) International Application Number: **PCT/US00/13724**
- (22) International Filing Date: **19 May 2000 (19.05.2000)** (81) Designated States (*national*): JP, KR.
- (25) Filing Language: **English** (84) Designated States (*regional*): European patent (AT, BE,  
CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC,  
NL, PT, SE).
- (26) Publication Language: **English**
- (30) Priority Data: **09/393,176** **9 September 1999 (09.09.1999)** **US** Published:  
— *With international search report.*
- (71) Applicant: **ADVANCED MICRO DEVICES, INC.**  
[US/US]; One AMD Place, Mail Stop 68, Sunnyvale, CA  
94088-3453 (US). *For two-letter codes and other abbreviations, refer to the "Guid-  
ance Notes on Codes and Abbreviations" appearing at the begin-  
ning of each regular issue of the PCT Gazette.*
- (72) Inventor: **MILLER, Michael, Lee**; 2614 Little Elm Trail,  
Cedar Park, TX 78613 (US).

(54) Title: **REAL-TIME FAULT DETECTION**



(57) Abstract: A method and apparatus for providing near real-time fault detection in a manufacturing process is provided. The apparatus includes a processing tool (105) adapted to manufacture a processing piece and an interface (110) coupled to the processing tool (105), for receiving operational data from the processing tool (105) related to the manufacture of the processing piece. In one embodiment, the processing tool (105) is in the form of semiconductor fabrication equipment and the processing piece is a silicon wafer. A fault detection unit (125) is provided to determine if a fault condition exists with the processing tool (105). An Advanced Process Control (APC) framework (120) is further provided to receive the operational data from the first interface (110), and to send the data to the fault detection unit (125) as the data is received by the first interface (110).

WO 01/18623 A1

## REAL-TIME FAULT DETECTION

### TECHNICAL FIELD

5 This invention relates generally to semiconductor fabrication technology, and, more particularly, to a method and apparatus for integrating near real-time fault detection capability into an Advanced Process Control (APC) framework.

### BACKGROUND ART

10 There is a constant drive in the semiconductor industry to increase the quality, reliability, and throughput of integrated circuit devices such as microprocessors, memory devices and the like. This drive is fueled by consumer demands for higher quality computers and electronic devices that operate more reliably.

These demands by the consumer have resulted in some improvements in the manufacture of semiconductor devices as well as in the manufacture of integrated circuit devices incorporating such semiconductor devices. Reducing the defects in the manufacture of these devices lowers the cost of the devices themselves. 15 Accordingly, the cost of the final product incorporating these devices is also reduced, thus providing inherent monetary benefits to both the consumer and manufacturer.

Although there has been an improvement in detecting faults associated with semiconductor manufacturing processes, one problem currently plaguing the semiconductor manufacturing industry is the delay in reporting these faults such that corrective measures can be implemented in a more expedient manner. As a result of this delay, 20 several faulty devices are produced, which undesirably increases costs for the manufacturer and consumer.

The present invention is directed to overcoming, or at least reducing the effects of, one or more of the problems set forth above.

### DISCLOSURE OF INVENTION

In one aspect of the present invention, a method is provided for fault detection in a manufacturing process. 25 The method comprises receiving operational data from a processing tool related to the manufacture of a processing piece at a first interface, sending the operational data from the first interface to a fault detection unit as the data is received by the first interface, and determining if a fault condition exists with the processing tool at the fault detection unit.

In another aspect of the present invention, a system is provided for fault detection in a manufacturing process. The system comprises a processing tool adapted to manufacture a processing piece. A first interface, 30 coupled to the processing tool, is adapted to receive operational data from the processing tool related to the manufacture of the processing piece. A fault detection unit is provided to determine if a fault condition exists with the processing tool. The system further includes a framework adapted to receive the operational data from the first interface, and to send the data to the fault detection unit as the data is received by the first interface.

### BRIEF DESCRIPTION OF THE DRAWINGS

35 The invention may be understood by reference to the following description taken in conjunction with the accompanying drawings, in which like reference numerals identify like elements, and in which:

Figure 1 illustrates a manufacturing system, including an APC framework, for providing near real-time fault detection of a processing tool in accordance with one embodiment;

40 Figure 2 depicts the detail of the APC framework of Figure 1; and

Figure 3 shows a process for providing fault detection in near real-time for the manufacturing system of Figure 1.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and are herein described in detail. It should be understood, however, that the description herein of specific embodiments is not intended to limit the invention to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

#### MODE(S) FOR CARRYING OUT THE INVENTION

Illustrative embodiments of the invention are described below. In the interest of clarity, not all features of an actual implementation are described in this specification. It will of course be appreciated that in the development of any such actual embodiment, numerous implementation-specific decisions must be made to achieve the developers' specific goals, such as compliance with system-related and business-related constraints, which will vary from one implementation to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking for those of ordinary skill in the art having the benefit of this disclosure.

Turning now to the drawings, and specifically referring to Figure 1, a system 100 for providing near real-time fault detection in a semiconductor fabrication process is provided. The system 100 includes a processing tool 105, which in the illustrated embodiment, is in the form of semiconductor fabrication equipment used to produce a processing piece, such as a silicon wafer, for example. It will be appreciated, however, that the processing tool 105 need not necessarily be limited to the production of silicon wafers, but could include other types of manufacturing equipment for producing a variety of different types of commercial products without departing from the spirit and scope of the present invention.

The processing tool 105 is coupled to an equipment interface (EI) 110, which retrieves various operational data from the tool 105, and communicates this data to an Advanced Process Control (APC) framework 120 to determine whether the tool 105 is experiencing a faulty operation. The equipment interface 110 further may receive control signals from the APC framework 120 that could be used to control the tool 105. For example, the control signal from the APC framework 120 could be used to shut down the tool 105 if the operational data that was sent by the equipment interface 110 was deemed faulty by the APC framework 120.

An add-on sensor 115 could also be coupled to the tool 105 to measure additional operational data that is not ascertained by the tool 105 itself. For example, the add-on sensor 115 could be used to determine whether the silicon wafer was produced within acceptable operational limits by the tool 105. Such acceptable operational limits of the tool 105 may be to produce the wafer within a certain temperature range, for example. It will be appreciated, however, that the add-on sensor 115 may be used to record various other operational parameters, and, thus, need not be limited to the aforementioned example.

The sensor 115 may be embodied as a simple data acquisition program, such as a C++ standalone program acquiring data from a thermocouple wire, for example. Alternatively, the sensor 115 may be embodied as a full-fledged LABVIEW application, acquiring data through multiple transducers (not shown). It will further be appreciated that the sensor 115 need not be used at all, and the APC framework 120 could rely solely upon the operational data forwarded from the equipment interface 110. If used, however, the sensor 115 forwards the additional operational data to the APC framework 120 for analysis.

A fault detection (FD) unit 125, which is coupled to the APC framework 120, receives the operational data of the tool 105 from the equipment interface 110 and sensor 115 via the framework 120. Prior to sending the operational data to the fault detection unit 125, however, the APC framework 120 translates the operational data to a format that is recognizable to the fault detection unit 125 in a manner that is well known to those of ordinary skill in the art. In accordance with one embodiment, the fault detection unit 125 includes a commercially available software package, such as ModelWare, for example, that provides fault detection analysis of the processing tool 105. It will be appreciated, however, that other types of commercially available fault detection software could also be used in lieu thereof without departing from the spirit and scope of the present invention.

The fault detection unit 125 compares the received operational data from the APC framework 120 to fault model data. The fault model data includes operational data of other similar-type tools, where it was previously known that such tools have operated within acceptable operational limits. The types of faults that could be detected by the fault detection unit 125 include processing and/or operational faults in silicon wafer fabrication. Examples of processing faults may include, but are not necessarily limited to, non-optimal preheating of the chamber, catastrophic failure where a broken wafer is detected, abnormal N<sub>2</sub> flow rate, temperature overshoots at the top of a ramp, tube temperature measurement drifts, *etc.* Some examples of operational faults detected by the fault detection unit 125 may include interrupted/resumed processing, no wafer sleuth or improper wafer sleuth prior to Rapid Thermal Anneal (RTA), *etc.*

The fault detection unit 125, upon evaluating the operational data sent from the APC framework 120, sends the results of potential faults and/or proper operation of the tool 105 to the APC framework 120. The APC framework 120, in turn, may send control signals to the equipment interface 110 to control the processing tool 105 based upon the results forwarded from the fault detection unit 125. For example, the control signal from the APC framework 120 may be to shut down the tool 105 to prevent any additional faulty production of wafers (providing this was determined by the fault detection unit 125). Data could also be sent from the APC framework 120 to inform a "fab" technician on how to rectify a faulty condition of the tool 105, if so desired.

Turning now to Figure 2, a more detailed representation of the APC framework 120 is provided. The APC framework 120 is a component-based architecture comprised of interchangeable, standardized software components enabling run-to-run control and fault detection of the processing tool 105. The APC framework 120 includes a machine interface (MI) 210 for communication with the tool 105 and the framework 120 to collect operational data therefrom. The APC framework 120 further includes a sensor interface (SI) 220 for communication between the add-on sensor 115 and the framework 120. The sensor interface 220 also collects operational data of the processing tool 105 through the sensor 115. A plan executor (PE) 230 (*i.e.*, a process controller) manages the APC framework 120 and provides possible solutions to problems found with the operational data that was determined by the fault detection unit 125. The framework 120 further includes an applications interface (AI) 240 for interfacing with third-party applications that run on the fault detection unit 125 to analyze the operational data received via the machine and sensor interfaces 210, 220. In the illustrated embodiment, the third-party application is the fault detection unit 125. A data channel 250 is further provided to allow for communication of data from the machine and sensor interfaces 210, 220, to the plan executor 230, and the applications interface 240 of the APC framework 120.

The machine interface (MI) 210 couples to the equipment interface 110 to serve as an interface between the processing tool 105 and the APC framework 120. The machine interface 210 supports the setup, activation, monitoring, and data collection of the tool 105. The machine interface 210 receives commands, status events, and

collected data from the equipment interface 110 and forwards this information to other components of the APC framework 120, namely the plan executor 230 and applications interface 240. Any responses that are received by the machine interface 210 from the other components of the APC framework 120 are routed to the equipment interface 110 for delivery to the processing tool 105. As previously discussed, this may include a control signal  
5 from the plan executor 230 to manipulate the tool 105 if a faulty condition is detected.

The machine interface 210 also reformats and restructures the messages between the specific communications protocol utilized by the equipment interface 110 and the Common Object Request Broker Architecture Interface Definition Language (CORBA IDL) communications protocol used by the components of the APC framework 120. The manner in which the machine interface 210 performs such translation between the  
10 equipment interface-specific communications protocol and the CORBA IDL protocol of the APC framework 120 is well known to those of ordinary skill in the art. Accordingly, the specific translation process between these two formats will not be discussed herein to avoid unnecessarily obscuring the present invention.

The sensor interface 220 couples the add-on sensor 115 to serve as an interface between the add-on sensor 115 and the APC framework 120. The sensor interface 220 provides setup, activation, monitoring, and data  
15 collection for the add-on sensor 115. Similar to the machine interface 210, the sensor interface 220 also reformats and restructures the messages between the specific communications protocol utilized by the sensor 115 and the CORBA IDL protocol used by the components of the APC framework 120.

The applications interface 240 supports the integration of third-party tools (e.g., commercial software packages, such as ModelWare, MatLab, and Mathematica, for example) to the APC framework 120. Typically,  
20 these third-party tools do not provide the standard CORBA IDL protocol known to the APC framework 120; accordingly, the applications interface 240 provides the necessary translation between the communications protocol utilized by the third-party tool and the CORBA protocol used by the APC framework 120.

In the illustrated embodiment, the third-party tool is the fault detection unit 125 for analyzing the operational data of the processing tool 105 that is supplied via the machine interface 210 and the sensor interface  
25 220. In one embodiment, the fault detection unit 125 includes ModelWare software for providing fault detection; however, it will be appreciated that other commercially available fault detection software could also be used without departing from the spirit and scope of the present invention.

The plan executor 230 performs control functions based upon the results determined by the fault detection unit 125. When the applications interface 240 receives the results from the fault detection unit 125, it forwards a  
30 copy of the results (usually in the form of an alarm signal) to the plan executor 230. Upon inspection of the results, the plan executor 230 will attempt to rectify any fault conditions with the tool 105 in a manner well known to those of ordinary skill in the art. The solution to a fault condition may be for the plan executor 230 to send a control signal to the machine interface 210 to shut down the tool 105 so as to prevent further manufacturing of faulty silicon wafers. The plan executor 230, in addition to shutting down the tool 105, may also apprise a "fab"  
35 technician of any potential solutions to rectify the fault condition through an operator interface (not shown), for example.

In a typical operation, the machine interface 210 and the sensor interface 220 usually forward the operational data obtained from the equipment interface 110 and sensor 115, respectively, to the plan executor 230. The plan executor 230 then buffers this operational data until a batch (*i.e.*, wafer-to-wafer or lot-to-lot) is  
40 completed by the processing tool 105. When the batch is complete, the plan executor 230 sends the accumulated operational data of the processing tool 105 to the applications interface 240, which then sends the data to the fault

detection unit 125. The fault detection unit 125 subsequently analyzes the received data and forwards the results back to the applications interface 240, which then forwards the results to the plan executor 230 for appropriate action.

A drawback with this typical operation, however, is that the results output from the fault detection unit 125 are usually determined after the batch is completed by the processing tool 105. Accordingly, the plan executor 230 cannot take immediate action to rectify the fault condition, and, thus, numerous faulty wafers could be produced as a result of this delay.

Turning now to Figure 3, a process 300 for the near real-time integration of fault detection in the APC framework 120 is provided. The process 300 commences at block 305, where the machine interface 210 and the sensor interface 220 receive operational data of the processing tool 105. In accordance with one embodiment, the machine interface 210 receives the operational data from the equipment interface 110, and the sensor interface 220 receives the operational data from the add-on sensor 115. In an alternative embodiment, the sensor 115 could be omitted, if so desired, in which case the operational data would then come solely from the equipment interface 110.

At block 310, the machine and sensor interfaces 210, 220 translate the operational data into a format that is recognizable to the plan executor 230 and application interface 240 of the APC framework 120 in a manner well established in the art. In accordance with one embodiment, this translation involves the reformatting and restructuring of messages between the specific communications protocol used by the equipment interface 110 and sensor 115 and the CORBA IDL protocol of the APC framework 120. Subsequent to receiving this translated data, the machine and sensor interfaces 210, 220 send the data via the data channel 250 to both the plan executor 230 and the applications interface 240 at block 315.

As the applications interface 240 receives the operational data in near real-time, it translates the data into a protocol used by the fault detection unit 125, and subsequently sends the data to the fault detection unit 125 at block 320. As previously discussed, the manner in which the applications interface 240 translates the data into the proper communications protocol is well known to those of ordinary skill in the art, and will differ depending on the particular type of fault detection software used. The fault detection unit 125, after receiving the operational data from the applications interface 240, compares the operational data to a fault model at block 325. As mentioned, the fault model includes operational data from other similar-type tools in which it was previously known that such tools manufactured silicon wafers within acceptable operational limits.

Subsequent to comparing the operational data of the tool 105 to the fault model data, the fault detection unit 125 sends the results of the comparison to the applications interface 240 at block 330. The applications interface 240 then translates the received results from the fault detection unit 125 into the CORBA IDL protocol used by the APC framework 120 at block 335. The applications interface 240 then forwards the results to the plan executor 230 at block 340, which is typically done in the form of an alarm signal. The plan executor 230, after receiving the alarm signal from the application interface 240, determines how to rectify the fault condition of the tool 105 at block 345 (providing that the tool 105 was actually deemed faulty). Rectifying the fault condition by the plan executor 230 may include a control signal being sent to the equipment interface 110 to shut down the tool 105, and to provide instructions to a "fab" technician on how to clear the fault, for example. The process in which the fault detection unit 125 determines how to rectify the fault condition is well within the knowledge of one of ordinary skill in the art. Accordingly, such process will not be discussed herein to avoid unnecessarily obscuring the present invention.

In accordance with the present invention, the operational data of the tool 105 is received in near real-time at the fault detection unit 250 before the batch processed by the tool 105 is complete. Accordingly, in contrast with typical fault reporting techniques, it is more likely that a fault will be cleared prior to the completion of the batch that is currently being processed by the tool 105.

- 5        The particular embodiments disclosed above are illustrative only, as the invention may be modified and practiced in different but equivalent manners apparent to those skilled in the art having the benefit of the teachings herein. Furthermore, no limitations are intended to the details of construction or design herein shown, other than as described in the claims below. It is therefore evident that the particular embodiments disclosed above may be altered or modified and all such variations are considered within the scope and spirit of the invention.
- 10        Accordingly, the protection sought herein is as set forth in the claims below.

## CLAIMS

1. A method for providing fault detection in a manufacturing process, comprising:  
receiving operational data from a processing tool (105) related to the manufacture of a processing piece at  
a first interface (110);  
5 sending the operational data from the first interface (110) to a fault detection unit (125) as the data is  
received by the first interface (110); and  
determining if a fault condition exists with the processing tool (105) at the fault detection unit (125).
2. The method of claim 1, wherein sending the operational data from the first interface (110) to a  
10 fault detection unit (125) further comprises:  
sending the operational data from the first interface (110) to a second interface (240) as the data is  
received at the first interface (110); and  
sending the operational data from the second interface (240) to the fault detection unit (125) as the data is  
received at the second interface (240).  
15
3. The method of claim 2, further comprising:  
sending the operational data from the first interface (110) to a process controller (230) as the data is  
received at the first interface (110).
- 20 4. The method of claim 3, further comprising:  
sending an alarm signal indicative of the fault condition to the process controller (230) from the second  
interface (240) providing that the fault condition was determined by the fault detection unit  
(125).
- 25 5. The method of claim 4, further comprising:  
performing a predetermined action to rectify the fault condition providing that the alarm signal is received  
by the process controller (230) from the second interface (240); and  
sending a control signal by the process controller (230) to the first interface (110) reflective of the  
predetermined action.  
30
6. A system for providing fault detection in a manufacturing process characterized in that said  
system comprises:  
a processing tool (105) adapted to manufacture to processing piece;  
a first interface (110) coupled to the processing tool (105), the first interface (110) adapted to receive  
35 operational data from the processing piece;  
a fault detection unit (125) adapted to determine if a fault condition exists with the processing tool (105);  
and  
a framework (120) adapted to receive the operational data from the first interface (110) and to send the  
data to the fault detection unit (125) as the data is received by the first interface (110).  
40



7. The system of claim 6, wherein the framework (120) includes:

a second interface (240), coupled to the fault detection unit (125), and adapted to receive the operational data as the data is received by the first interface (110) during the manufacture of the processing piece, and to send the operational data to the fault detection unit (125) as the data is received at the second interface (240).

8. The system of claim 7, wherein the framework further includes:

a process controller (230) coupled to the first and second interfaces (110, 240), the process controller (230) to receive the operational data as the data is received at the first interface (110) during the manufacture of the processing piece.

9. The system of claim 8, wherein the second interface (240) is further adapted to send an alarm signal to the process controller (230) providing that a fault condition was determined by the fault detection unit (125), and wherein the process controller (230) is further adapted to perform a predetermined action to rectify the fault condition providing the alarm signal is received, and to send a control signal to the first interface (110) reflective of the predetermined action.

10. The system of claim 6, wherein the framework (120) further includes:

a third interface (210), coupled between the first interface (110), the process controller (230), and the second interface (240), the third interface (210), adapted to receive the operational data from the first interface (110) and to translate the operational data between a first communications protocol used by the first interface (110) and a second communications protocol used by the framework (120).

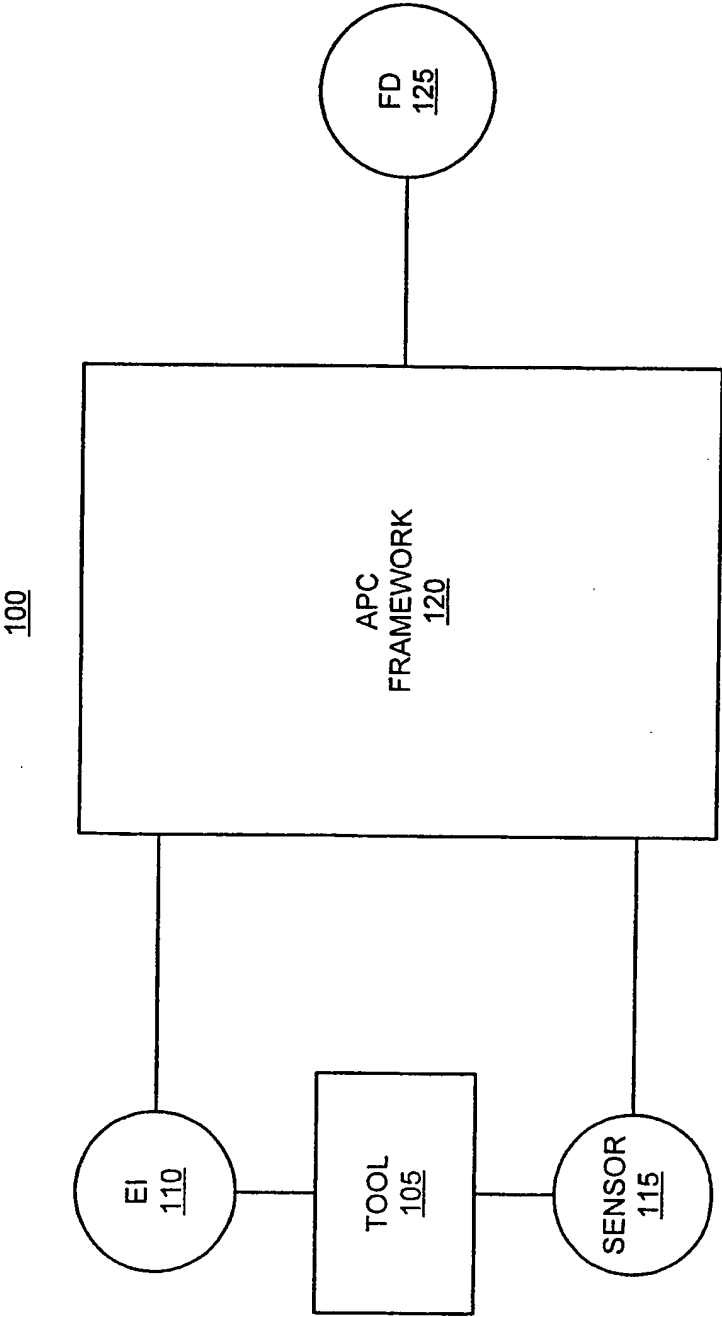


Figure 1

2 / 4

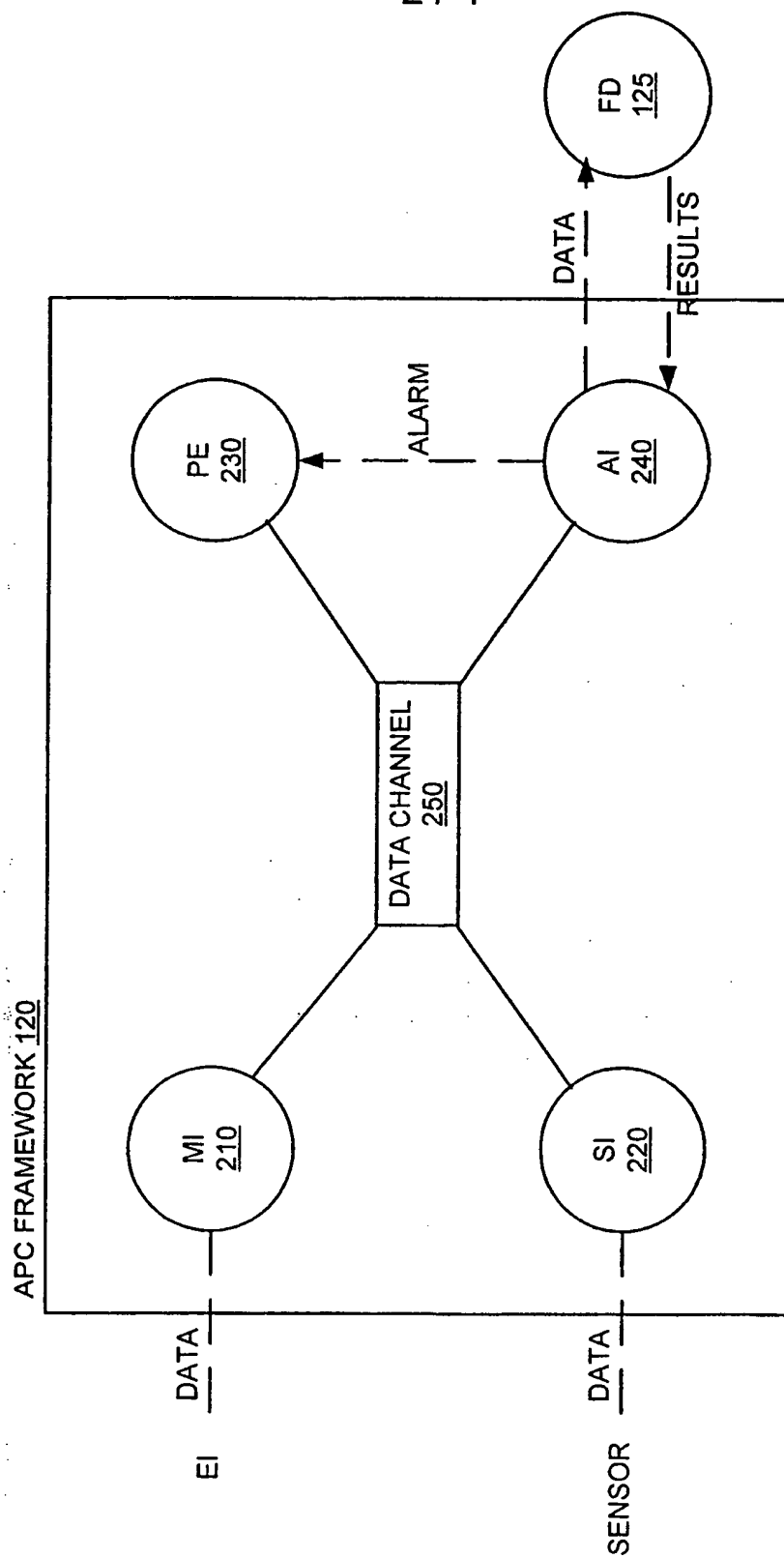
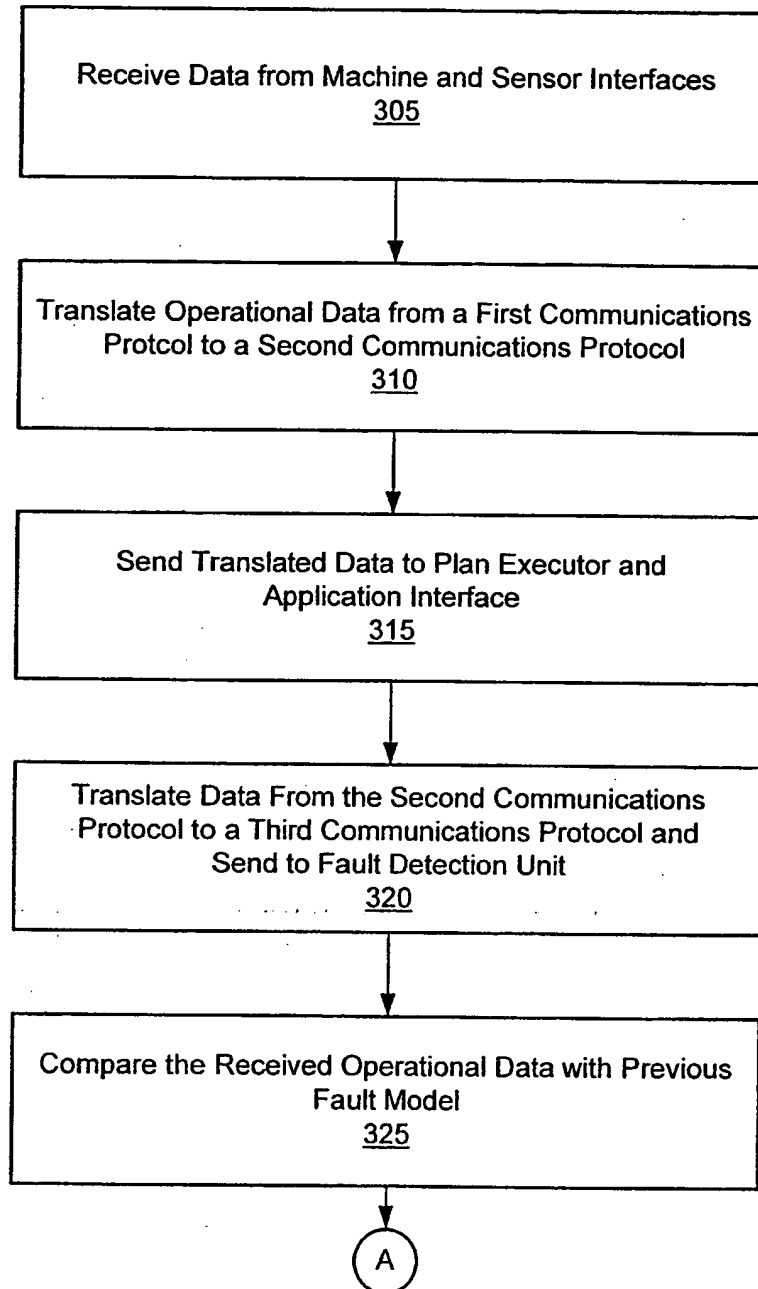
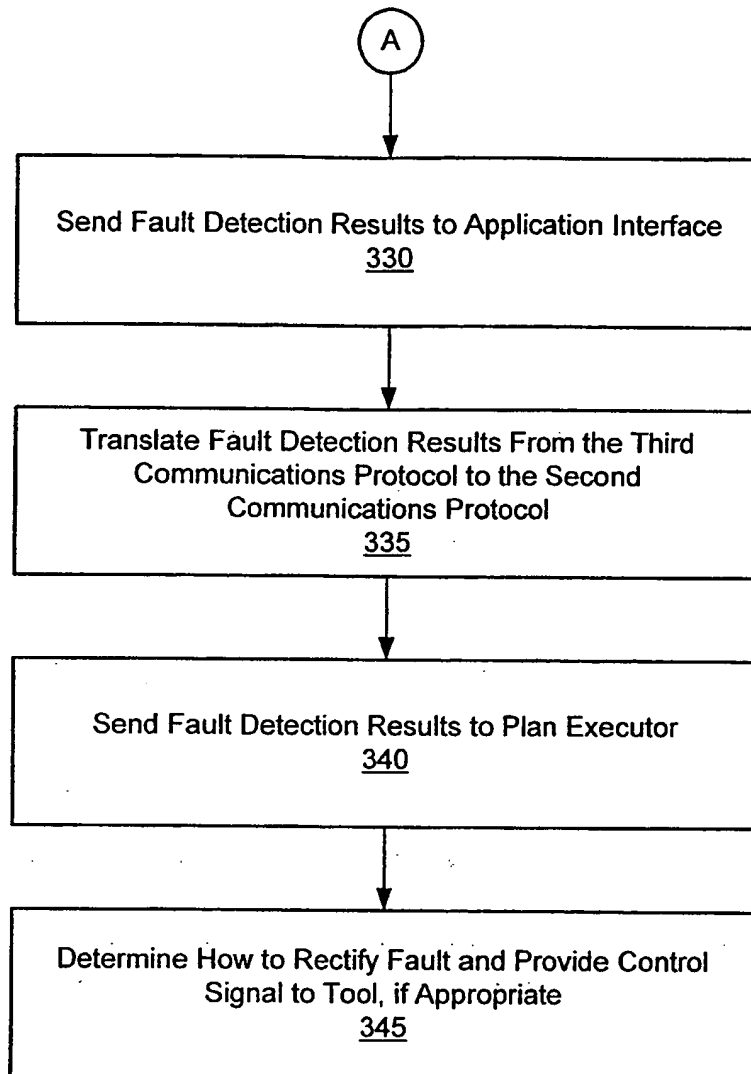


Figure 2

3 / 4

300**Figure 3A**

4 / 4

**Figure 3B**

# INTERNATIONAL SEARCH REPORT

International Application No <b>PCT/US 00/13724</b>		
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC 7 G05B19/418		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) IPC 7 G05B		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, WPI Data		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 847 529 A (HSIEH C R ET AL) 8 December 1998 (1998-12-08) column 2, line 28 -column 3, line 36; figures 2,4	1,6
X	US 5 661 669 A (MOZUMDER PURNENDU KANTI ET AL) 26 August 1997 (1997-08-26) column 5, line 38 -column 6, line 29; claim 1; figures 1,2	1,6
X	WO 99 16108 A (ON LINE TECHN INC) 1 April 1999 (1999-04-01) page 2, line 10 -page 3, line 29; figure 1	1,6
A	US 5 740 062 A (BERKEN LLOYD M ET AL) 14 April 1998 (1998-04-14) abstract	1-10
-/-		
<div style="display: flex; justify-content: space-between;"> <span><input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.</span> <span><input checked="" type="checkbox"/> Patent family members are listed in annex.</span> </div>		
* Special categories of cited documents : <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </div> <div style="width: 45%;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&amp;" document member of the same patent family</p> </div> </div>		
Date of the actual completion of the international search <div style="text-align: center; font-weight: bold;">29 August 2000</div>		Date of mailing of the international search report <div style="text-align: center; font-weight: bold;">06/09/2000</div>
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer <div style="text-align: center; font-weight: bold;">Tran-Tien, T</div>

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/13724

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>US 5 859 964 A (WANG QINGSU ET AL)  12 January 1999 (1999-01-12)  abstract</p> <p>-----</p>	1-10

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/13724

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5847529 A	08-12-1998	NONE	
US 5661669 A	26-08-1997	US 5526293 A	11-06-1996
WO 9916108 A	01-04-1999	AU 9501998 A	12-04-1999
US 5740062 A	14-04-1998	US 5563798 A	08-10-1996
US 5859964 A	12-01-1999	NONE	